

I CLAIM:

1 1. A method for testing changes in a software program using a plurality
2 of test cases, wherein the software program comprises a first plurality of execution
3 paths, the method comprising:

4 identifying one or more changed paths in the first plurality of execution
5 paths;

6 from the plurality of test cases, identifying one or more test cases that
7 are capable of executing the one or more changed paths; and

8 executing the one or more of the identified test cases to test the changed
9 path.

1 2. The method of claim 1, wherein the software program comprises one
2 or more modules, and identifying one or more test cases comprises identifying a
3 changed module and determining whether the changed module causes changes in
4 the execution paths.

1 3. The method of claim 1, wherein identifying one or more test cases
2 comprises identifying a second plurality of execution paths in the software
3 program and determining the difference between the first and second pluralities of
4 execution paths.

1 4. The method of claim 3, wherein the difference comprises at least one

2 of a new path and a changed path.

1 5. The method of claim 1, wherein identifying one or more test cases
2 comprises evaluating names of one or more methods of a test case from the
3 plurality of test cases thereby determining whether the methods of the test case
4 involve the one or more changed paths.

1 6. The method of claim 5, wherein identifying one or more test cases
2 further comprises evaluating parameters of one or more methods of a test case
3 from the plurality of test cases thereby determining whether the methods of the
4 test case involve the one or more changed paths.

1 7. The method of claim 1, wherein identifying one or more test cases
2 comprises determining whether a test case intersect one or more changed paths.

1 8. The method of claim 7, wherein determining whether a test case
2 intersect one or more changed execution paths comprises identifying a module of
3 the software program included in both the test case and a changed execution path.

1 9. The method of claim 8, wherein each module is represented by a
2 node number, and each execution path and test case is represented by a string of
3 node numbers, wherein identifying a module comprises identifying a node number
4 included in both a changed execution path and a test case.

1 10. A computer program product for testing a software program using a
2 plurality of test cases, the computer program product comprising a computer
3 usable medium having a computer readable program code embodied thereon, the
4 computer readable program code controlling the computer to perform the
5 operations of:

6 identifying one or more changed paths in a first plurality of execution
7 paths of the software program;

8 identifying one or more test cases that are capable of executing the one
9 or more changed paths; and

10 executing the identified one or more test cases to test the changed code
11 of the software program.

1 11. The computer program product of claim 10, wherein the software
2 program comprises one or more modules, wherein identifying one or more paths
3 comprises identifying the changed module and determining whether the changed
4 module causes changes in the execution paths.

1 12. The computer program product of claim 10, wherein identifying one
2 or more paths comprises identifying a second plurality of execution paths in the
3 software program upon changing of the code and determining the difference
4 between the first and second pluralities of execution paths.

1 13. The computer program product of claim 12, wherein the difference

2 comprises at least one of a new path and a changed path.

1 14. The computer program product of claim 10, wherein identifying one
2 or more test cases comprises evaluating the names of one or more methods of a
3 test case from the plurality of test cases thereby determining whether the methods
4 of the test case involves the one or more changed paths.

1 15. The computer program product of claim 14, wherein identifying one
2 or more test cases further comprises evaluating the parameters of one or more
3 methods of a test case from the plurality of test cases thereby determining whether
4 the methods of the test case involve the one or more changed paths.

1 16. The computer program of claim 10, wherein identifying one or more
2 test cases comprises determining whether a test case intersects one or more
3 changed paths.

1 17. The computer program of claim 16, wherein determining whether a
2 test case intersect one or more changed execution paths comprises identifying a
3 module of the software program included in both the test case and a changed
4 execution path.

1 18. The computer program of claim 17, wherein each module is
2 represented by a node number, and each execution path and test case is
3 represented by a string of node numbers, wherein identifying a module comprises
4 identifying a node number included in both a changed execution path and a test

5 case.

1 19. A system for testing changes in a software program using a plurality
2 of test cases, wherein the software program comprises a first plurality of execution
3 paths, the system comprising:

4 means for identifying one or more changed paths in the first plurality of
5 execution paths;

6 means for identifying one or more test cases from the plurality of test
7 cases that are capable of executing the one or more changed
8 paths,

9 wherein the one or more identified test cases are executed to test the
10 changed code of the software program.

1 20. The system of claim 19, wherein the software program comprises
2 one or more modules, wherein upon changing of the code at least one module is
3 changed, and wherein identifying one or more test cases comprises identifying the
4 changed module and determining whether the changed module causes changes in
5 the execution paths.

1 21. The system of claim 19, wherein identifying one or more test cases
2 comprises identifying a second plurality of execution paths in the software
3 program upon changing of the code and determining the difference between the
4 first and second pluralities of execution paths.

1 22. The system of claim 21, wherein the difference comprises at least
2 one of a new path and a changed path.

1 23. The system of claim 19, wherein identifying one or more test cases
2 comprises evaluating names of one or more methods of a test case from the
3 plurality of test cases thereby determining whether the methods of the test case
4 involve the one or more changed paths.

1 24. The system of claim 23, wherein identifying one or more test cases
2 further comprises evaluating the parameters of one or more methods of a test case
3 from the plurality of test cases thereby determining whether the methods of the
4 test case involve the one or more changed paths.

1 25. The system of claim 19, wherein identifying one or more test cases
2 comprises determining whether a test case intersects one or more changed paths.

1 26. The system of claim 25, wherein determining whether a test case
2 intersect one or more changed execution paths comprises identifying a module of
3 the software program included in both the test case and a changed execution path.

1 27. The system of claim 26, wherein each module is represented by a
2 node number, and each execution path and test case is represented by a string of
3 node numbers, wherein identifying a module comprises identifying a node number
4 included in both a changed execution path and a test case.